

- $I_L("s" \hat{=} \text{rdf:XMLLiteral}) \notin LV$  und
- $\langle I_L("s" \hat{=} \text{rdf:XMLLiteral}), \text{rdf:XMLLiteral}^I \rangle \notin I_{\text{EXT}}(\text{rdf:type}^I)$ .

Zusätzlich zu diesen semantischen Einschränkungen stellen wir für RDF-Interpretationen die Forderung, dass sämtliche der nachfolgenden Tripel (genannt *axiomatische RDF-Tripel*) als wahr ausgewertet werden müssen:

<code>rdf:type</code>	<code>rdf:type</code>	<code>rdf:Property</code>	.
<code>rdf:subject</code>	<code>rdf:type</code>	<code>rdf:Property</code>	.
<code>rdf:predicate</code>	<code>rdf:type</code>	<code>rdf:Property</code>	.
<code>rdf:object</code>	<code>rdf:type</code>	<code>rdf:Property</code>	.
<code>rdf:first</code>	<code>rdf:type</code>	<code>rdf:Property</code>	.
<code>rdf:rest</code>	<code>rdf:type</code>	<code>rdf:Property</code>	.
<code>rdf:value</code>	<code>rdf:type</code>	<code>rdf:Property</code>	.
<code>rdf:_1</code>	<code>rdf:type</code>	<code>rdf:Property</code>	.
<code>rdf:_2</code>	<code>rdf:type</code>	<code>rdf:Property</code>	.
...			
<code>rdf:nil</code>	<code>rdf:type</code>	<code>rdf:List</code>	.

Bis auf das letzte dienen alle dieser Tripel dazu, Ressourcen, die speziellen RDF-URIs zugeordnet sind, als Property zu kennzeichnen. Dies erfolgt durch die Typisierung als `rdf:type rdf:Property`, die aufgrund der obigen Definition von RDF-Interpretationen genau die gewünschten semantischen Auswirkungen hat.

Gemeinsam stellen diese Restriktionen sicher, dass sich eine RDF-Interpretation entsprechend der beabsichtigten Bedeutung verhält.

Ganz analog zur Definition der einfachen Folgerung sagt man nun, ein Graph  $G_2$  *RDF-folgt* aus einem Graphen  $G_1$ , wenn jede RDF-Interpretation, die Modell von  $G_1$  ist, auch ein Modell von  $G_2$  ist.

### ➤ 4.2.3 RDFS-Interpretationen

Wie in Abschnitt 3.4 dargestellt, reichert RDFS das RDF-Vokabular um weitere speziell zu interpretierende Konstrukte an. Beispielsweise werden (analog zu `rdf:Property`) neue Klassenbezeichner eingeführt, die es (via `rdf:type`) ermöglichen, eine URI explizit als Repräsentant für eine Ressource, als ungetyptes Literal oder selbst als Klassenbezeichner zu kennzeichnen. Neue Property-URIs ermöglichen wiederum, den Gegenstands- und Wertebereich von Property durch Klassenbezeichner zu charakterisieren sowie Klassen- wie Propertybezeichner zueinander in hierarchische Beziehung zu setzen, womit es möglich wird, Schema- bzw. terminologisches Wissen in Form von Tripeln auszudrücken.

Das speziell zu interpretierende RDFS-Vokabular  $V_{\text{RDFS}}$  besteht aus folgenden Bezeichnern:

`rdfs:domain` `rdfs:range` `rdfs:Resource` `rdfs:Literal` `rdfs:Datatype`